



HashiTalks



VaultPass: Enabling Teams to Share Secrets Confidentially

Browser extension for the Vault Key-Value store




Chris Blum



Passionate Hashicorp SW user
he/him

 @ChrisNBlum

 mulbc



01 WHY • HOW • DEMO • FUTURE

WHY VaultPass was created

What was the pain?



I used to work for a hosting company



My team had a lot of credentials, some changed on a schedule

- To collect the secrets, they used KeePassXC
 - Free, available on most systems, Browser Extensions exist
- To synchronize, they used Git
 - Well understood in the team, free, easy to set up, undelete possible
 - Works better than a shared drive
- To secure they added GPG encryption
 - Everyone in the team was able to decrypt what was in git

This created problems



- KeePassXC
 - Some secrets had to be shared with other teams, opening multiple DBs not ~~possible~~ convenient
 - Every Ops team had access to ALL secrets

This created problems



- Git
 - The encrypted KeePassXC DB is treated as a binary
 - Changes in binaries are not easy to compact
 - Git was growing, but ok for now
 - No way to know when Git had to be pulled until a password failed
 - Good luck merging your updates with upstream changes

This created problems



- GPG
 - No way to revoke access to secrets
 - They are on the local disk after all...
 - Especially problematic since passwords changed infrequently



This appears to be common in companies

(and works surprisingly well until you rotate passwords more frequently)

— 02 WHY • **HOW** • DEMO • FUTURE

HOW VaultPass was created

What's the architecture?



How VaultPass was created



- Centrally managing secrets in a safe way is difficult
 - Let's leave that to other people
- Hashicorp Vault

How VaultPass was created



- Handling secrets for other people requires trust
 - Small & Easy Open Source codebase (in plain JavaScript)
 - ONE external library: [WebExtension browser API Polyfill](#) by Mozilla
 - Easy to review by everyone
- Became so easy, Hashicorp [made a tutorial](#) based on the codebase



Store stats

- Chrome Store
 - 417 current users (30% growth in 6 months)
 - Users from the US (80), Germany (58), France (40), Russia (30), ...
- Firefox store
 - 47 current users (34% growth in 6 months)
 - Most users use Windows (47%) followed by Linux (38%), Mac at 15%

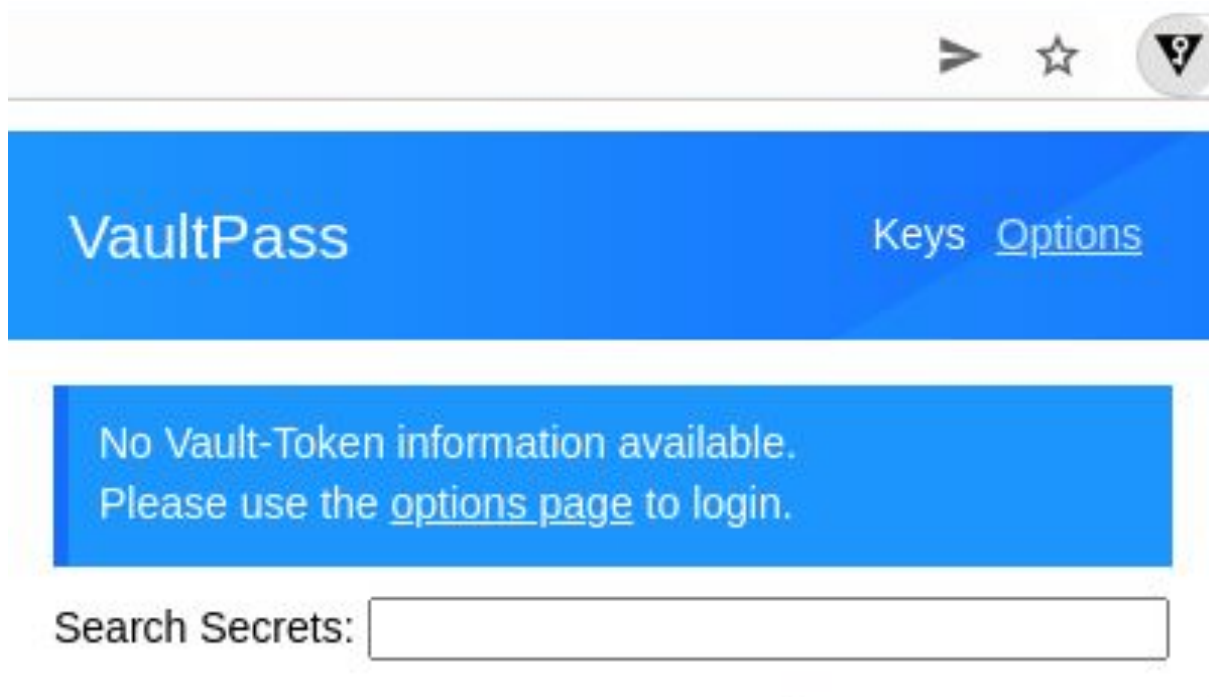
03 WHY • HOW • **DEMO** • FUTURE

DEMO VaultPass

Let me get a look (finally)



Our journey starts here



Login through the options menu



The screenshot shows a browser window with the VaultPass login page. The page has a blue header with the text "VaultPass" and a link for "Options". Below the header, there are four input fields: "Vault server URL" with the value "http://localhost:8200", "Username" with the value "mitchellh", "Password" with masked characters "***", and "Auth Mountpoint" with the value "userpass". At the bottom, there is a blue button labeled "Login to Vault".

VaultPass [Keys](#) [Options](#)

Vault server URL:

Username:

Password:

Auth Mountpoint:

[Login to Vault](#)

See all credential organisations



The screenshot shows a web browser window with a blue header bar. The header contains the text "VaultPass" on the left and "[Keys](#) [Options](#)" on the right. Below the header is a blue box with the text "Attached policies:" followed by a list of policies: "admins" and "default". Below this box, there are two entries, each with a key icon and a label: "admin/" and "denied/". Each entry has an "Active" status with a checkbox. The "admin/" entry has a checked checkbox, while the "denied/" entry has an unchecked checkbox. At the bottom of the interface is a "Logout" button.

VaultPass [Keys](#) [Options](#)

Attached policies: [×](#)

- admins
- default

admin/ Active

denied/ Active

Logout

Create credentials in an org (admin)



The screenshot shows the Vault interface for creating a credential. The top navigation bar includes a dropdown menu, 'Secrets', 'Access', 'Policies', and 'Tools'. The breadcrumb trail is '< secret < vaultPass < admin < google.com'. The main heading is 'vaultPass/admin/google.com'. Below this, there are three tabs: 'JSON' (disabled), 'Org' (selected), and 'URL regex'. The 'Org' tab is highlighted with a light blue callout. The 'URL regex' tab is also highlighted with a light blue callout. Below the tabs is a table with two columns: 'Key' and 'Value'. The table contains two rows: one for 'password' with a value of 'unsafe' and one for 'username' with a value of 'testUser'. Each row has a copy icon and an eye icon to its right.

Key	Value
password	unsafe
username	testUser

Setup KV v2 Policies



• • • /sys/policy/default

CODE EDITOR

```
# Allow listing all orgs in VaultPass
path "secret/metadata/vaultPass" {
  capabilities = [
    "list",
  ]
}

# Deny any access to vaultPass credentials by default
path "secret/data/vaultPass/*" {
  capabilities = [
    "deny",
  ]
}
```

Setup KV v2 Policies



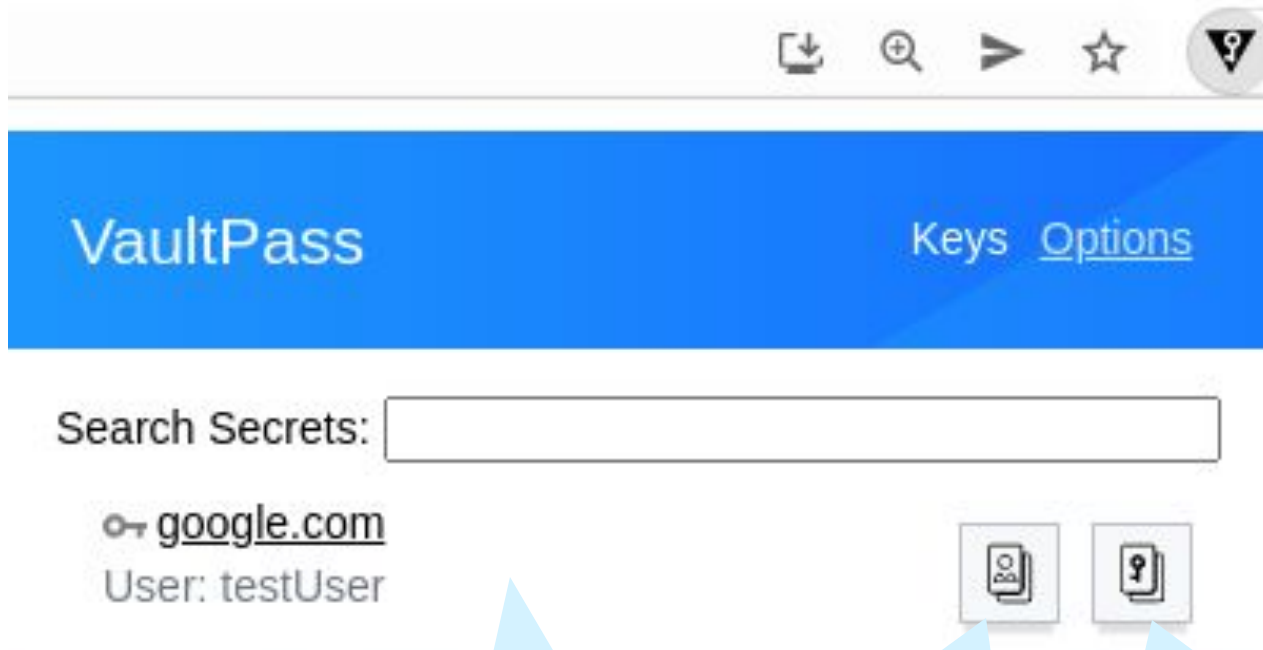
• • • /sys/policy/admin

CODE EDITOR

```
# Allow admins to list secrets in their org
path "secret/metadata/vaultPass/admin/" {
  capabilities = [
    "list",
  ]
}

# Allow admin full access to their credentials
path "secret/data/vaultPass/admin/*" {
  capabilities = [
    "create",
    "read",
    "update",
    "delete",
  ]
}
```

Use credentials on a site



Click fills login prompt

Copy Username

Copy Password

— 04 WHY • HOW • DEMO • **FUTURE**

Future of VaultPass

Next steps...



Pending work



- Make it easier to get started with VaultPass
 - Most people struggle with Vault policies
- SSO login for Vault
- Creating & Updating secrets through VaultPass
- KeePass DB migration script
- Better way to match secrets with URLs?



Thank You

hugs@hashicorp.com | learn.hashicorp.com | discuss.hashicorp.com